IN THE SPECIFICATION

Please amend the paragraph at page 3, lines 14-25, as follows:

--In accordance with a third aspect of the invention, there is disclosed a computer program product having a computer readable medium having a computer program recorded therein for entropy coding of discrete wavelet transform (DWT) coefficient bits that are arranged into code blocks and coded in bitplane order using three coding passes for each bitplane. The computer program product includes:

a computer program code module for pre-analyzing transform coefficients of a code block in sign-magnitude form;

a computer program code means for storing statistical data about the coefficients; and

a computer program code means for, based upon the statistical data, generating at least one command for at least one sequence of bit and context pairs for arithmetic encoding.--

Please amend the paragraph beginning at page 4, line 31, and ending at page 5, line 7, as follows:

--In accordance with a sixth aspect of the invention, there is disclosed a computer program product having a computer readable medium having a computer

program recorded therein for entropy coding of discrete wavelet transform (DWT) coefficient bits that are arranged into code blocks and coded in bitplane order using three coding passes for each bitplane.[[.]] The computer program product includes: a computer program code module for generating a context for each bit of one or more coefficients in a code block; a computer program code module for arithmetic coding each bit to be coded from the code block using the context for the bit; and a computer program code module for providing a FIFO between the context generating and the arithmetic coding to streamline transfer of data between the context generator and the arithmetic coder, the FIFO adapted to store each bit, the corresponding context and a repeat number of the bit and context pair.--

Please amend the paragraph at page 12, lines 15-23, as follows:

--Once the highest significant bit-plane is coded by the Cleanup process 516, the rest of the bit-planes are coded by a Code1Plane process. In particular, in decision block 518, a check is made to determine if N>0 (indicating bitplanes remain to be processed). If decision block 518 returns false (N), processing terminates in step 520. Otherwise, if decision block 518 returns true (Y), processing continues at step 522. In step 522, the number of significant bitplanes is decremented. In step 524, the Code1Plane (N) process is carried out (see Fig. 6). Processing then continues at decision block 518. The output of each pass is a sequence of bits and their corresponding contexts. These bit and context ~~pair~~ pairs are written into a FIFO 120 for use by the arithmetic coder 130.--

Please amend the paragraph at page 21, lines 1-15, as follows:

--The coefficient memory module 1140 has the same structure as the coefficient memory 1000 shown in Fig. 10, in the register-array-based method. The coefficients are pre-analysed in the fashion described in Section 2.7, and the statistics are stored together along with the coefficients in the coefficient memory 1140. Instead of caching the whole SigState, Coded and MR1st bit planes in register arrays, a small register-window for each is used to cache the region being coded, while the other regions that are not being coded are kept in the scratch memory. Preferably, a region is confined to 8 columns in a scan. After the context of one of the 3 passes for a particular region is generated, the updated state of SigState, Coded and MR1st bits is written back to the scratch memory 1110. The corresponding SigMatrix bit is also updated with the result of ORing all the bits of SigState in a region, on the write back to the ScratchMem module. Each entry of the scratch memory 1110 holds the state of SigState, Coded and MR1st data for a region. A 2-port (1R/1W) memory is preferably used as the scratch memory 1110 to improve efficiency.--